

Examining Collaboration in interdisciplinary Product Development focusing on Dependencies

Guenther Schuh
WZL, RWTH Aachen
University
g.schuh@wzl.rwth-aachen.de

Michael Riesener
WZL, RWTH Aachen
University
m.riesener@wzl.rwth-aachen.de

Christian Mattern
WZL, RWTH Aachen
University
c.mattern@wzl.rwth-aachen.de

Abstract

Product development in manufacturing industry is characterized by intense collaboration need of various stakeholders. Increasing integration of disciplines in modern products makes it more and more a challenge to arrange collaboration efficiently and effectively. Process and product characteristics as well as the architecture of information systems used in product development have to be considered.

This paper introduces a methodology for the design of collaboration situations based on principles of system analysis. First, a collaboration situation is defined and modelled regarding constituent elements in the domains process, product and system. Second, a description model for dependencies in these domains is developed. Morphological analysis was applied to derive features and characteristics of the model. Third, an improvement approach to optimize a given collaboration situation is depicted. The improvement approach comprises a sensitivity model, which explicates causal relations between the dependency features. The methodology is applied to a case study from manufacturing industry.

when engineers from different disciplines must work together for joint solution finding, specific shortcomings occur. For example, relevant dependencies between components of the product are not considered appropriately resulting in late changes in the development process [3]. Additionally, separate IT systems used in the different disciplines are not adequately compatible. Although suitable solutions are available, the selection and the directed use of the suitable systems remains a challenge. Recent activities focus on accessibility of relevant information across the whole lifecycle of products and platforms [4]. Thus, there is a relevant potential for increase of effectiveness and efficiency in product development when collaboration between disciplines is designed appropriately [5].

This paper proposes a methodology in order to address the described challenges. A feature-based description model is developed using morphological analysis which allows to abstract a given collaboration situation. The application of the description model makes the collaboration situation analyzable so that further methods can be applied. In order to evaluate the modelled collaboration situation, an approach based on a sensitivity model is presented, which contains causal relations within the collaboration situation.

1. Introduction

Interdisciplinary collaboration is gaining more and more importance in the product development in manufacturing industry. Whereas mechanical engineering traditionally had a dominant role, today's products consist of an increasing share of electrics/electronics and software [1]. Due to recent developments in the context of "Industrie 4.0" regarding the trend towards connection of mechatronic products to cyber-physical systems, interdisciplinarity is a substantial characteristic of product development across all sectors [2].

However, collaboration across disciplines leads to specific challenges in product development. Discussions with practitioners show that, especially

2. Related research

2.1. Design of collaboration in product development

An established process model for interdisciplinary development of mechatronic products is provided by VDI 2206, which includes the V-model comprising the phases system design, discipline-specific design and system integration [6]. The work of EIGNER regarding Model-Based Systems Engineering (MBSE) focuses on the development of a comprehensive system model which integrates different disciplines with their perspective on the product and their individual information structure regarding product-related

information [7]. One of the overarching objectives of the MBSE approach is to overcome the barriers between the disciplines due to low interoperability of IT-systems.

Collaboration Engineering as an area of research analyzes the collaboration between different stakeholders in general and offers guidance regarding the usage of IT-tools. The approach of KOLFSCHOTEN AND VREEDE, for example, aims at supporting the design of repeatable collaboration processes [8]. A selection of collaboration tools for specific collaboration situations is supported by the CSCW-Matrix from JOHANSEN [9].

Mutual relations between IT architecture and organizations are subject to the work of MACCORMACK ET AL. [10]. BLOOM ET AL. investigate the impact of information technology on the organizational structure of companies [11]. SCHUH ET AL. provide a framework for analysis of the impact which technological advancements related to Industrie 4.0 have on collaboration productivity [12, 13]. However, the focus in these research contributions is not on product development specifically.

2.2. Dependency analysis based on Design Structure Matrices

Dependency analysis is a widely recognized instrument for analysis of structures in product development regarding the product structure just as communication structures among engineering teams [14].

Following this understanding, a dependency is a one-sided or mutual relationship among elements of a system. These elements are usually classified into domains according to LINDEMANN [14]. Much work in this area has been conducted with the help of the Design Structure Matrix (DSM), e.g. by PIMMLER, EPPINGER and SOSA [15, 16]. TRISTL addresses the challenges of collaboration between the disciplines systems engineering and mechanical/electrical engineering with help of Multiple Domain Matrices (MDM) [17]. The approach of HELLENBRAND intends to support the synchronization of the disciplines with an MDM-based system model, as well [18].

The DSM-based approaches facilitate the computing of acquired data and thus automated application. Most of the approaches are limited to a binary description of dependencies though.

2.3. Differentiated dependency analysis

Other authors also investigate dependencies in product development, but pursue more differentiated approaches regarding the description of dependencies.

The contributions of MOSER ET AL. [19] and CHUCHOŁOWSKI ET AL. [20] include approaches to analyze dependency mechanisms and to identify characteristics for evaluation of dependencies in product development projects with focus on dependencies between activities. GIACHETTI provides a method to describe dependencies between partial systems of enterprises and differentiates between types of dependencies [21]. GROSSMANN ET AL. analyze generic approaches to model dependencies within business processes and propose an extension of the Unified Modeling Language (UML) [22]. KERN provides a feature-based approach to describe dependencies with focus on distributed product development [23].

The analysis of the above-mentioned approaches shows that a more differentiated analysis of dependencies has been conducted regarding specific application areas. Some authors use features or characteristics to describe the dependencies. However, a comprehensive analysis of dependencies in product development has not been conducted so far.

2.3. Summary and research gap

The discussion of the related research shows that collaboration in enterprises is subject to many studies. However, only few contributions address collaboration in product development in manufacturing industry specifically. Considering the economic relevance, it is an important field of research though. A large research stream analyzes product development focusing on dependencies with matrix-based methods. There are only a few approaches, which allow a more differentiated analysis as discussed in the previous chapter. None of these approaches uses the dependency-focused view on product development for analysis and improvement of collaboration. The approach presented in this paper aims at filling this research gap.

3. Methodology for the design of collaboration situations in interdisciplinary product development

3.1. Description model for collaboration situations

Basis for the analysis and design of dependencies in product development is a comprehensive description of the relevant field of observation. For this purpose, the description model for collaboration situations in interdisciplinary product development is subdivided into three domains: the process domain, the product

domain and the system domain. These domains cover the relevant fields of actions in product development formulated by EHRENSPIEL: Development process, technical-economical issues and organizational issues including the organization of IT-tools [24].

Each of the domains is modeled with elements and corresponding dependencies. The process domain consists of activities representing elements and interactions between them representing dependencies. In the product domain, the elements are represented by architecture elements including corresponding requirements, functions and product components. The dependencies in the product domain result from mutual dependence between the architecture elements regarding physical, functional or requirement-related dependence. The system domain consists then of components from the IT architecture and their couplings. The domains as well as the elements and the dependencies are shown in Figure 1.

In order to make the interdisciplinary product development accessible for further analysis, a description model has been developed, which represents the core elements and their relations.

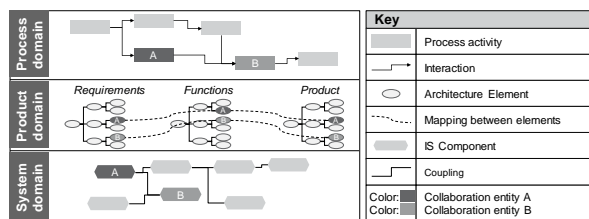


Figure 1. Collaboration situations in interdisciplinary product development

The description model characterizes a collaboration situation as shown in Figure 2. The collaboration situation is subdivided into the three domains process, product and system. Each collaboration situation consists of two combined collaboration entities. Each collaboration entity is constituted by an activity in the process domain and dedicated elements in the two other domains. In the product domain, the architecture element, which is treated within the activity, is part of the corresponding collaboration entity. In the system domain, the system, which is used to perform the work within the activity, is part of the collaboration entity. This approach of describing a collaboration situation follows the principles of Structural Complexity Management where dependencies between multiple domains are analyzed in Multiple Domain Matrices (MDM) [14]. In case there are multiple architecture elements treated or systems used, the primarily treated or used ones are selected as part of the collaboration entity.

For the relations of the elements within each domain it is assumed that there is a dependency between the elements, which can be described using features and characteristics.

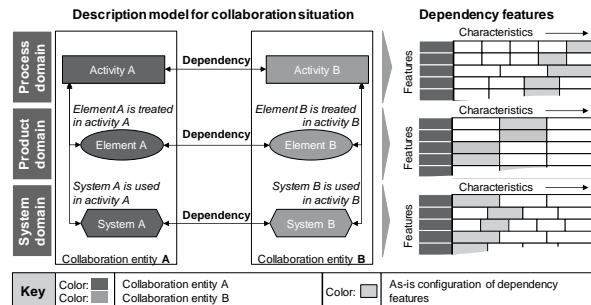


Figure 2. Description model of a collaboration situation

Because the analysis of the collaboration situation requires a differentiated perspective on the dependencies between the collaboration entities, a morphological analysis was conducted to identify the relevant features and corresponding characteristics for description of the dependencies. These features and characteristics were integrated into morphological boxes. The morphological boxes allow to describe a single dependency by configuration of characteristics in each domain. A detailed description of the identified features is part of the following chapter.

3.2. Morphological analysis of dependencies

The features in the morphological boxes are structured in superordinate categories, which were derived initially from existing literature in adjacent fields as presented in chapter 2. To ensure they are exhaustive, they were complemented by help of reflection of industrial cases, where relevant data was acquired and discussions with experts were conducted.

Table 1 shows the categories and features for the process domain. The categories are derived from a general conception of business processes according to which they are constituted of elements (activities), relations (input-output-relations) and their temporal sequence [25]. The categories cover aspects regarding time (for the constituent “temporal sequence”), direction, content, form, distance (each for the constituent “relations”) as well as participants and disciplines (each for the constituent “elements”) involved. For each feature in the categories, individual characteristics were defined, which are shown in the third column of the table. For reasons of clarity, only the extremal values are shown here. The lowest value within the range is considered as the characteristic for the lowest dependency regarding the corresponding

feature. The highest value shown in the table represents the characteristic where the highest dependency regarding the corresponding feature is assumed. As a result, the characteristics of the lowest value would be assigned to a dependency between activities of short duration and low frequency. On the other hand, a dependency with long duration and high frequency would be evaluated with the characteristics of the highest value, which are located on the right side of the morphological box.

Table 1. Dependency features for process domain

Category	Features	Range of characteristics
Goal	Goal of interaction	information handover – collaborative problem solving
Time	Length of interaction	short – unlimited
	Frequency of interaction	one-time – continuous
	Time span	high distance – parallelism
Direction	Direction of information flow	one-sided – mutual
Content	Content richness	low – high
	Content complexity	low – high
	Content extent	low – high
	Degree of abstraction	low – high
Form	Formalization	low – high
	Standardization	low – high
	Documentation	low – high
	Intensity	low – high
Distance	Spatial distance	high – low
	Organizational distance	high – low
	Cultural distance	high – low
Partici-pants	Number of participants	1:1 – n:n
Disci-plines	Hierarchy of participants	different –equal
	Diverseness of disciplines	not related – equal
	Product perspective	not transferable – equal
	Mutual comprehension	low – high

The categorization for the dependency features in the product domain as shown in Table 2 is based on the general domains for product models, which include

requirements, functions and the product structure [26]. The categories *Requirements* and *Functions* include features regarding the common fulfillment, the exclusiveness of fulfillment and the mutual relation of the requirements or functions of the considered architecture elements. The features in the category *Product* include the mechanical coupling as well as the flow of material, information and energy. An additional category regarding product program-related dependencies was added, which describes dependencies between different products of a product program that are interlinked e.g. due to standardization.

Table 2. Dependency features for product domain

Category	Features	Range of characteristics
Require-ments	Common fulfillment	none – completely
	Exclusiveness of fulfillment	none – completely
	Mutual relation	none – completely
Functions	Common fulfillment	none – completely
	Exclusiveness of fulfillment	none – completely
	Mutual relation	none – completely
	Process-oriented relation	none – completely
	Hierarchical relation	none – completely
Product	Mechanical coupling	none – mutual
	Material flow	none – mutual
	Information flow	none – mutual
	Energy flow	none – mutual
	Packaging constraints	none – mutual
	Standardization	low – high
Product program	Program-related relation	none – full

The dependency features in the system domain as shown in Table 3 describe the coupling between elements of the information system architecture, which are equivalent to authoring systems in most cases. On the upper level, four categories were defined that cover the type of dependency between the systems, the user interface, the kind of information transfer as well as the data structure. These categories cover all relevant aspects regarding information technology which is used as a tool in product development. [27] Especially the information transfer and the data structure are from particular importance when exchange of information

between engineers of different disciplines is needed. In current industry application the tools e.g. used by mechanical and electrical engineering show problematic media discontinuities and available solutions in the field of model-based engineering are not established yet [7].

Table 3. Dependency features for system domain

Category	Features	Range of characteristics
Type	Homogeneity	low – high
	Functional conformity	low – high
	Functional scope	completely different – equal
User interface	Interface conformity	low – high
Information transfer	Automation of transfer	low – high
	Frequency of transfer	no synchronization – live
	Latency of transfer	high – none
	Variety of medium	high – low
	Homogeneity of medium	low – high
	Property of medium	third party – own
	Property of medium	third party – own
Data structure	Unity of structure	low – high
	Editability	low – high
	Ambiguity	high – low
	Codification	low – high

3.3. Assessment and improvement of a collaboration situation

The afore explained description model allows to describe dependencies in the three domains with the defined features and characteristics. In order to assess a collaboration situation regarding the design of the dependencies, a guideline is required whether the dependencies on each domain are suitably configured in terms of the features. A suitable configuration of dependencies means, that there is a fit between the dependencies on each domain. A dependency in the product domain should be mirrored by appropriate dependencies in the process and the system domain. In case there is no fit between the dependencies in the three domains, specific measures for design of the dependencies have to be taken.

In a first step, it is assumed that the right configuration of the dependencies can be reduced to generic causal relations between the dependency features. There are three overall types of causal relations as shown in Figure 3:

- The type *Prerequisite* means that a given rating in feature A requires a specific rating in feature B.
- The type *Consequence* means that a given rating in feature A leads to a specific rating in feature B.
- The type *Joint occurrence* means, that a given rating in feature A comes along with a specific rating in feature B.

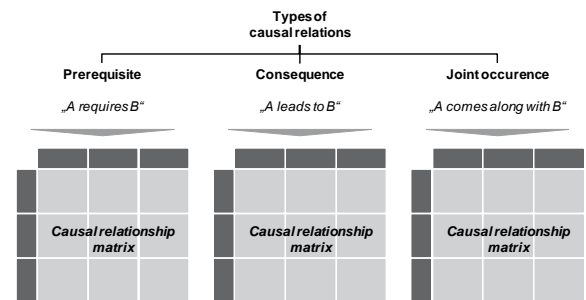


Figure 3. Types of causal relations

For the three types, the causal chains are defined for combinations of two features. In the following, causal relations from type “Prerequisite” will be discussed in more detail.

In general, there are some universal statements, which can be made regarding the causal chains from the type “Prerequisite”:

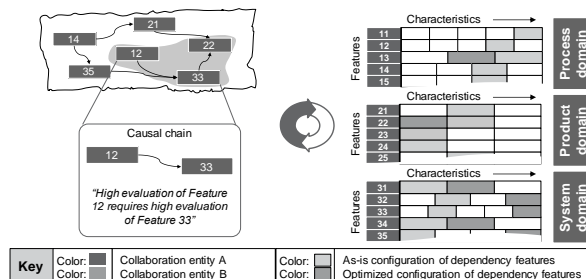
- A high evaluation regarding the features of a dependency in the process domain usually requires a high evaluation in the system domain.
- A high evaluation regarding the features of a dependency in the product domain requires a high evaluation in the product and system domain only in specific features.
- Given evaluations of dependency features in the system domain only require specific evaluations in other domains as the given evaluations are low
- Given evaluations of dependency features in the process domain only require specific evaluations in the product domain as the given evaluations are low

Figure 4 shows examples for causal relations from type *Prerequisite* as verbal descriptions for selected dependency features in each domain.

	Process domain	Product domain	System domain
Process domain	Time Content A high content-wise dependency between activities requires a continuous interaction and partial parallelism.	Product Time A dependency between activities with high time span and discrete interactions requires low dependency within the product regarding physical coupling and flows.	Information transfer Time A high time-wise dependency between activities requires a highly automated information sharing with high frequency and low latency.
Product domain	Time Requirements A high requirements-based dependency between architecture elements requires a long, one-time or repeated interaction; a temporal distance is feasible.	Feature: Functions Requirements A high requirements-based dependency requires a high function-based dependency as functions need to fulfill the defined requirements.	Information transfer Product A high dependency between architecture elements regarding physical coupling and flows requires high automation and frequency as well as low latency of information sharing.
System domain	Time Information transfer A low homogeneity between authoring systems with low frequency and high latency requires short and discrete interactions with low intensity.	Product Information transfer A low homogeneity between authoring systems with low frequency and high latency requires low coupling and flows between architecture elements.	Data structure Information transfer A high dependency between authoring systems regarding information sharing requires a common database with high editability and low ambiguity.

Figure 4. Examples for causal relations from type *Prerequisite*

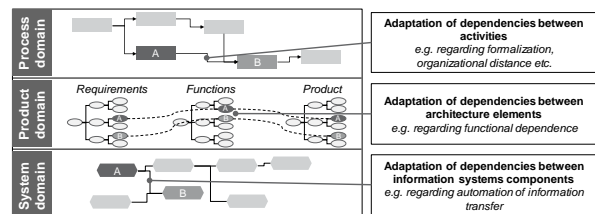
The different causal chains between the dependency features are integrated into a sensitivity model according to VESTER [28] as shown in Figure 5. The sensitivity model allows to analyze multilateral relations between the dependency features as well as second order relations. In the context of this approach, the sensitivity model serves as a reference for the design of collaboration situations. By extracting sub-models from the overall sensitivity model, guidelines for efficient and effective design of collaboration situations can be derived.



Based upon the evaluation with the help of the description model, the collaboration situation is assessed and improved. The evaluation of the collaboration situation in the morphological boxes of the description model is compared to the causal relations in the sensitivity model. In case the evaluation does not fit with the guidelines from the sensitivity model, a preferred evaluation of the corresponding features is derived. To overcome existing differences between the evaluated as-is collaboration situation and the to-be situation according to the preferred evaluation, specific improvement measures need to be taken. On the right side, Figure 6 shows some

examples for measures for adaptation of dependencies in the three domains.

Depending on the affected dependency feature, individual adaptation measures e.g. regarding the formalization of the interaction or the organizational distance can be taken in the process domain. In the product domain, measures regarding the structure of the product architecture on requirements, functional or product structure level are possible to either increase or decrease the dependency between the architecture elements that are part of the considered collaboration situation. In the system domain, measures for adaptation of the information transfer between the used systems can be taken to adapt the coupling between the regarded systems e.g. in terms of automation or frequency.



4. Case study

4.1. Introduction to the case study

The methodology was applied to a collaboration situation in a manufacturing company. Essential steps presented in the previous chapter are demonstrated using this example.

The case is taken out of a process analysis of the development process in the field of ventilation technology. A specific step in the development process is considered, where the design of the casing is evaluated using a molding simulation.

This is a typical example where two different departments with individual expertise and thus from different disciplines need to collaborate. The mechanical engineer develops a concept for the plastic casing of a new fan. The casing needs to be suitable for injection molding. Therefore, the simulation expert analyzes the concept using moldflow analysis and identifies weak points in the current design. Both engineers work together to improve the design.

In the given case example, the involved employees felt, that collaboration was not conducted efficiently. However, the problem areas were unclear and improvement potentials were not clearly identifiable.

In order to analyze and improve the collaboration situation, the developed methodology was applied to the case. First, the situation was described using the

concept of the “collaboration situation” as introduced in this paper. Second, the description model was used to model the dependencies in the given case. Third, improvements were derived.

4.2. Application of the methodology

The collaboration situation with two collaboration entities is structured as follows. In the first collaboration entity there is the activity “Design of casing” in the system domain, the architecture element “Casing” in the product domain and the corresponding CAD tool, which is used for definition of the casing geometry, in the system domain.

The second collaboration entity includes the activity “Simulate molding” in the process domain and the corresponding CFD tool, which is used for the simulation in the system domain. In the product domain, the same architecture element is included as in the first collaboration entity, because both parties of the collaboration entity work on the same element of the product. Figure 7 gives an overview of the collaboration situation.

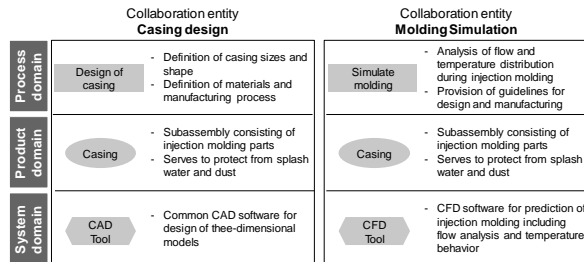


Figure 7. Overview of collaboration situation in case study

In the given collaboration situation, there are dependencies between the elements in the three domains. The dependencies in the case example were evaluated using the description model. In the following, some selected features of particular importance will be described.

The goal of the interaction is to exchange information between the two collaboration entities. The length of the interaction is considered to be “medium-long” as there are only discrete interactions regarding the simulation task, the approach and the results. As a result, the frequency of interaction is evaluated as “multiple interactions”. However, the documentation is high because the results of the simulation must be documented due to regulatory conditions.

The dependency features in the product domain are rated at the highest level because both collaboration entities work on the same architecture element of the product.

In the system domain it must be considered, that the required information for the molding simulation is transferred via a PowerPoint form. This means that the engineer occupied with the design of the casing has to extract the relevant data from the 3D model and entry the data into that form manually. On the other side, the simulation expert has to extract the data again from the PowerPoint form and entry the data into the simulation program manually. Thus, the evaluation of the corresponding dependency features “Automation of transfer”, “Latency of transfer”, “Unity of structure” and “Ambiguity” are low.

A short verbal description of the dependencies and the rating regarding the previously mentioned dependency features is shown in Figure 8.

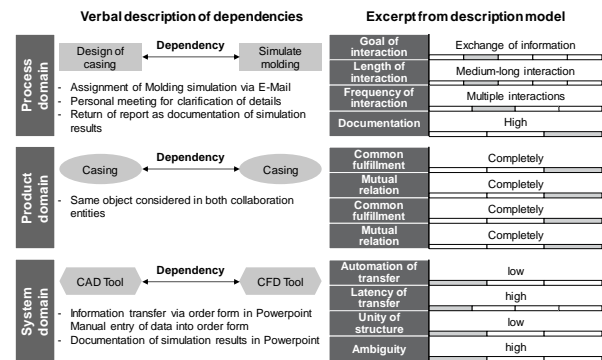


Figure 8. Application of description model to the collaboration situation in case study

After the application of the description model for the given collaboration situation, the rated dependency features were compared to the causal relations in the sensitivity model. As a result, deviations from the guidelines given by the sensitivity model were derived.

The first deviation is between the feature “Documentation” in the process domain and the feature “Ambiguity” in the system domain. The requirements regarding documentation of the transferred information between the two collaboration entities are not reflected in the degree of ambiguity in the process domain. As the data has to be transferred between different forms that follow the individual nomenclature of the involved disciplines, ambiguity is high. This and two further deviations, which were derived, are shown in Figure 9.

According to the identified deviations, an improved configuration of the dependency features was derived. In order to meet the requirements regarding the automation, unity of the data structure and ambiguity, the corresponding evaluation of the dependency features was changed. The improved configuration of the dependency features defines how the collaboration situation should be improved in order to increase efficiency and effectiveness.

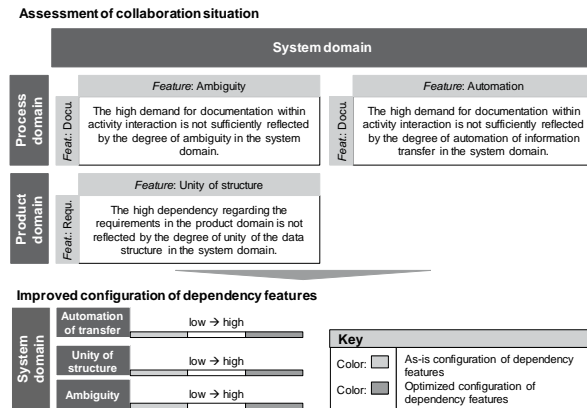


Figure 9. Assessment and improvement of collaboration situation

As a result, the dependency in the case study between the two collaboration entities in the system domain was improved through implementation of an Engineering Data Management Tool, which allows to automatically transfer the relevant data between the collaboration entities. Thus, the application of the model helped to improve the collaboration between casing design and molding simulation with specific measures.

5. Conclusion

The methodology introduced in this paper enables a design of various collaboration situations in interdisciplinary product development regarding the three domains process, product and system. In a first step, the description model for a collaboration situation consisting of two collaboration entities was presented. The description model comprises morphological boxes, which were introduced in a second step. The morphological boxes contain features and characteristics for the evaluation of dependencies between elements of the collaboration situation in the three domains. In a third step an approach for assessment of the evaluated collaboration situation with the help of a sensitivity model was explained. The sensitivity model provides causal relations between the dependency features. Types and examples for the causal relations were explained in this paper. Based on the causal relations and improved configuration of the dependency features for the collaboration situation can be derived. The methodology was applied to a development process in the field of ventilation technology.

The presented methodology contributes from academic perspective to existing studies on dependencies in product development. Furthermore, it offers insights into the specifics of collaboration in

product development. Practitioners benefit from the results by a systematic approach to improve collaboration situations. However, limitations of the methodology result from the qualitative approach, which was selected to identify and describe dependencies and causal relations.

Future work will focus on the empirical foundation of the identified causal relations and the quantification of the effects from improvement measures in the three domains on collaboration productivity.

6. References

- [1] R. Isermann, *Mechatronische Systeme: Grundlagen*, 2nd ed. Berlin, Heidelberg, New York: Springer, 2008.
- [2] M. Abramovici and O. Herzog, Eds., *Engineering im Umfeld von Industrie 4.0: Einschätzungen und Handlungsbedarf*. München: Herbert Utz, 2016.
- [3] G. Schuh, *Lean Innovation*. Berlin, Heidelberg: Springer Vieweg, 2013.
- [4] G. Schuh, M. Riesener, and E. Schrey, "Development of an Information Model for Lifecycle Management of Modular Product Platforms," in *Proceedings of the ISPIM Innovation Forum - Fostering Innovation Ecosystems*, Toronto, 2017.
- [5] G. Baum, "Innovationen als Basis der nächsten Industrierevolution," in *Industrie 4.0: Beherrschung der industriellen Komplexität mit SysLM*, U. Sendler, Ed., Berlin, Heidelberg: Springer Vieweg, 2013, pp. 37–53.
- [6] *Entwicklungsmethodik für mechatronische Systeme*, VDI 2206, 2004.
- [7] M. Eigner, D. Roubanov, and R. Zafirov, Eds., *Modellbasierte virtuelle Produktentwicklung*. Berlin, Heidelberg: Springer Vieweg, 2014.
- [8] G. L. Kolfshoten and G.-J. de Vreede, "A Design Approach for Collaboration Processes: A multimethod design science study in collaboration engineering," *Journal of management information systems*, vol. 26, no. 1, pp. 225–256, 2009.
- [9] R. Johansen, *Groupware: Computer support for business teams*. New York: The Free Press, 1988.
- [10] A. MacCormack, R. Lagerstrom, D. Dreyfus, and C. Baldwin, "Building the Agile Enterprise: IT Architecture, Modularity and the Cost of IT Change," *Harvard Business School Working Paper 15-060*, 2016.
- [11] N. Bloom, L. Garicano, R. Sadun, and J. van Reenen, "The distinct effects of information technology and communication technology on firm organization," *Management Science*, vol. 60, no. 12, pp. 2859–2885, 2014.

- [12] G. Schuh, T. Potente, R. Varandani, C. Hausberg, and B. Fränken, "Collaboration Moves Productivity To The Next Level," *Procedia CIRP*, no. 17, pp. 3–8, 2014.
- [13] G. Schuh, T. Potente, C. Wesch-Potente, A. R. Weber, and J.-P. Prote, "Collaboration Mechanisms to increase Productivity in the Context of Industrie 4.0," *Procedia CIRP*, no. 19, pp. 51–56, 2014.
- [14] U. Lindemann, M. Maurer, and T. Braun, *Structural Complexity Management: An Approach for the Field of Product Design*. Berlin, Heidelberg: Springer, 2009.
- [15] T. U. Pimpler and S. D. Eppinger, "Integration analysis of product decompositions," 1994.
- [16] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "The misalignment of product architecture and organizational structure in complex product development," *Management Science*, vol. 50, no. 12, pp. 1674–1689, 2004.
- [17] C. Tristl, A. Karcher, H. Klenk, and C. Haubach-Lippmann, "Towards a Framework for Synchronization of Systems- and Mechanical/Electrical Engineering processes on multiple dimensions," in *Proceedings of the 19th ISPE International Conference on Concurrent Engineering*, Trier, 2012.
- [18] D. Hellenbrand, "Transdisziplinäre Planung und Synchronisation mechatronischer Produktentwicklungsprozesse," *Dissertation, Lehrstuhl für Produktentwicklung*, TU München, München, 2013.
- [19] B. Moser, W. Grossmann, and P. Starke, "Mechanisms of Dependence in Engineering Projects as Sociotechnical Systems," *Transdisciplinary Lifecycle Analysis of Systems*, no. 2, pp. 142–151, 2015.
- [20] N. Chucholowski, Starke, Phillip, B. Moser, and U. Lindemann, "Characterizing and Measuring Activity Dependence in Engineering Projects," in *Proceedings of the Portland International Conference on Management of Engineering & Technology 2016*, Honolulu, Hawaii, USA, 2016.
- [21] R. Giachetti, "Understanding Interdependence in Enterprise Systems: A Model and Measurement Formalism," in *Proceedings of the International Conference on Business Process Management*, 2006, pp. 257–267.
- [22] G. Grossmann, M. Schrefl, and M. Stumptner, "Modelling Inter-Process Dependencies with High-Level Business Process Modelling Languages," *Proceedings of the 5th Asia-Pacific conference on Conceptual Modeling*, vol. 79, pp. 89–102, 2008.
- [23] E.-M. Kern, *Verteilte Produktentwicklung: Rahmenkonzept und Vorgehensweise zur organisatorischen Gestaltung*. Berlin: GITO, 2005.
- [24] K. Ehrlenspiel and H. Meerkamm, *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit*, 5th ed. München: Carl Hanser, 2013.
- [25] H. J. Schmelzer and W. Sesselmann, *Geschäftsprozessmanagement in der Praxis: Kunden zufriedenstellen, Produktivität steigern, Wert erhöhen : [das Standardwerk]*, 8th ed. München: Hanser, 2013.
- [26] U. Lindemann, *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden*, 3rd ed. Dodrecht, Heidelberg, London, New York: Springer, 2009.
- [27] M. Eigner, F. Gerhardt, T. Gilz, and F. M. Nem, *Informationstechnologie für Ingenieure*. Berlin, Heidelberg: Springer-Verlag, 2012.
- [28] F. Vester, *Die Kunst vernetzt zu denken: Ideen und Werkzeuge für einen neuen Umgang mit Komplexität; ein Bericht an den Club of Rome*, 3rd ed. München: Deutscher Taschenbuchverlag, 2003.